

A decorative vertical bar on the left side of the slide, consisting of a wide light gray section and a narrower dark gray section.

STRINGS

PUC Minas
Michelle Nery Nascimento
ATP

Strings

- C# oferece suporte interno para um tipo string totalmente funcional
- C# trata strings como objetos que encapsulam todos os métodos de manipulação, classificação e busca
- Em C#, string não é um array de caracteres

Strings

- Uma string é uma série de caracteres tratados como uma única unidade
- Uma string é um objeto da classe String no espaço de nomes System
- Escrevemos literais de string ou constantes de string (strings literais) como sequência de caracteres entre aspas duplas

“José P. Souza”

“Rua Principal, 9999”

“Belo Horizonte – MG”

- Uma declaração pode atribuir uma **string** literal a uma referência a **string**

string cor = “azul”;

- Inicializa a referência a string cor para se referir ao objeto literal “azul”
- C# fornece a palavra-chave **string** como um alias da classe **String**

Construtores de strings

- A classe String fornece 8 construtores para inicializar strings de várias maneiras

```
class ConstroiString
```

```
{
```

```
    public static void Main (string[] args)
```

```
    {
```

```
        string stringOriginal, string1, string2, string3, string4, string5, string6;
```

```
        char [ ] VetorDeCaracter = { 'F' , 'e' , 'l' , 'i' , 'z' , ',' , 'd' , 'i' , 'a' };
```

```
        stringOriginal = "Bem vindo";
```

```
        string1 = stringOriginal;
```

```
        string2 = new string (VetorDeCaracter);
```

```
        string3 = new string (VetorDeCaracter, 6, 3);
```

```
        string4 = new string ( 'C', 5);
```

```
        string5 = "Linha 1\nLinha 2";
```

```
        string6 = @"Linha1
```

```
Linha2";
```

```
        Console.WriteLine("String1: {0} \nString2: {1} \nString3: {2} \nString4: {3}
```

```
\nString5: {4} \nString6:{5} ", string1, string2, string3, string4,  
string5, string6);
```

```
    }
```

Construtores de strings

- Saída

```
String1: Bem vindo
String2: Feliz,dia
String3: dia
String4: CCCCC
String5: Linha 1
Linha 2
String6:Linha1
Linha2
```

```
Press any key to continue...■
```

ToString()

- Outra maneira de se criar uma string é chamando o método ToString() em um objeto e atribuindo o resultado a uma variável string

```
int meuInteiro = 5;  
string inteiroParaString = meuInteiro.ToString();  
Console.WriteLine("Inteiro para string: " + inteiroParaString;
```

Saída

Inteiro para string: 5

Strings

- CopyTo: Copia um número especificado de caracteres de uma posição especificada nesta instância para uma posição especificada em um array de caracteres

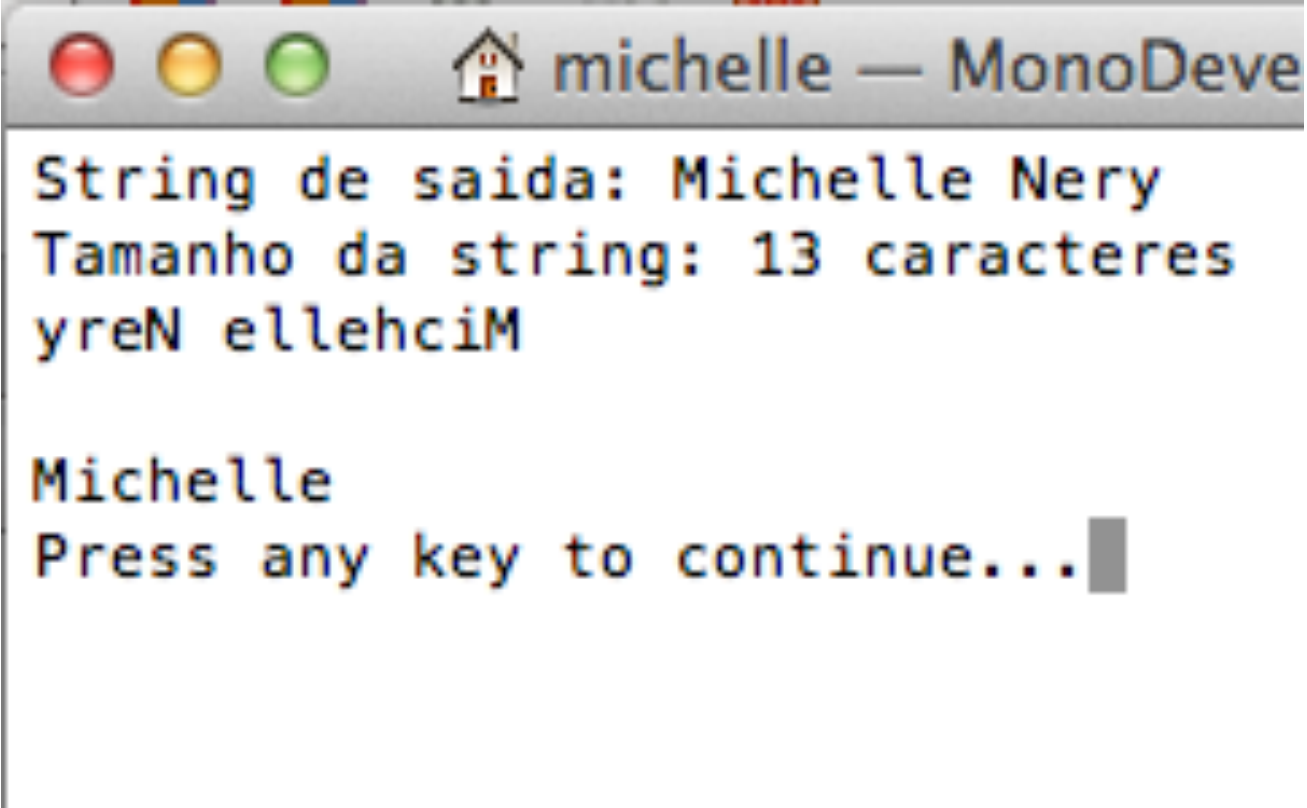

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1;
        char [ ] vetorDeCaracter = new char [13];

        string1 = "Michelle Nery";
        Console.WriteLine (" String de saida: " + string1);
        Console.WriteLine ("Tamanho da string: {0} caracteres", string1.Length);

        //percorre os caracteres da string e exibe-os de maneira invertida
        for (int i = string1.Length -1; i>=0; i--)
            Console.Write (string1 [i]);
        Console.WriteLine("\n");

        //copia caracteres da string para o vetor de caracteres
        string1.CopyTo (0, vetorDeCaracter, 0, 8);
        for (int i = 0; i <vetorDeCaracter.Length; i++) {
            Console.Write (vetorDeCaracter [i]);
        }
    }
}
```

- Saída



The image shows a screenshot of a terminal window titled "michelle — MonoDeve". The window has standard macOS window controls (red, yellow, green buttons) and a home icon. The terminal displays the following text:

```
String de saida: Michelle Nery  
Tamanho da string: 13 caracteres  
yreN ellehciM  
  
Michelle  
Press any key to continue...
```

The text is displayed in a monospaced font. The output shows the string "Michelle Nery", its length of 13 characters, and its reverse "M yreN ellehciM". It then prints "Michelle" and prompts the user to "Press any key to continue..." with a cursor.

Comparando Strings

- O método **Equals** testa a igualdade de qualquer dois objetos (verifica se os objetos possuem conteúdos idênticos)
- O método retorna **true** se os objetos são iguais e **false**, caso contrário
- **Equals** usa uma *comparação lexicográfica* – os valores Unicode inteiros que representam cada caracter são comparados

Comparando Strings

- Método Equals

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1 = "Ola";
        Console.WriteLine ("String 1: {0}" , string1);
        //testa a igualdade usando metodo Equals
        if(string1.Equals ("Ola"))
            Console.WriteLine ("A string 1: {0} eh igual a Ola");
        else
            Console.WriteLine ("A string 1: {0} nao eh igual a Ola");
    } }
```

Comparando Strings

- Saída

```
String 1: Ola
```

```
A string 1: {0} eh igual a Ola
```

```
Press any key to continue...
```

Comparando Strings

- Em C#, o operador de igualdade (==) também utiliza uma comparação lexicográfica para comparar duas strings

Comparando Strings

- Operador de igualdade (==)

```
class ConstroiString
```

```
{
```

```
    public static void Main (string[] args)
```

```
    {
```

```
        string string1 = "Ola";
```

```
        Console.WriteLine ("String 1: {0}" , string1);
```

```
        //testa a igualdade usando operador de igualdade
```

```
        if(string1 == "OLA")
```

```
            Console.WriteLine ("A string 1: {0} eh igual a OLA");
```

```
        else
```

```
            Console.WriteLine ("A string 1: {0} nao eh igual a OLA");
```

```
    } }
```

Comparando Strings

- Saída

```
String 1: Ola
```

```
A string 1: {0} nao eh igual a OLA
```

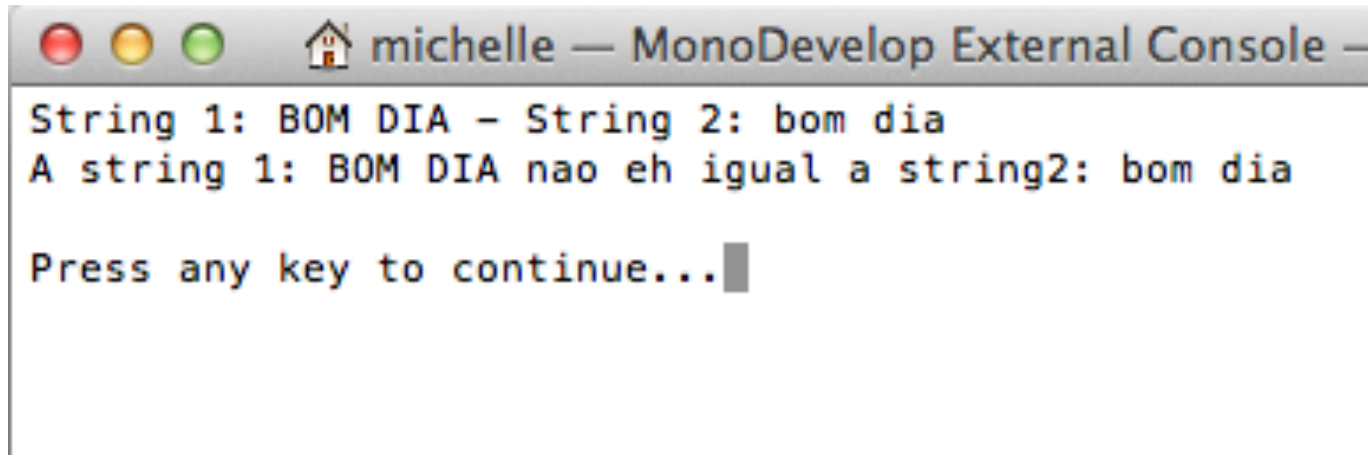
```
Press any key to continue...
```


Comparando Strings

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1 = "BOM DIA";
        string string2 = "bom dia";

        Console.WriteLine ("String 1: {0} - String 2: {1}" , string1, string2);
        //testa a igualdade usando método Equals
        if (string1.Equals(string2))
            Console.WriteLine ("A string 1: {0} eh igual a string 2: {1}", string1, string2);
        else
            Console.WriteLine ("A string 1: {0} nao eh igual a string2: {1}", string1,
string2);
    }
}
```

- Saída



```
michelle — MonoDevelop External Console —  
String 1: BOM DIA - String 2: bom dia  
A string 1: BOM DIA nao eh igual a string2: bom dia  
Press any key to continue...
```

Comparando Strings

- A comparação entre a string1 – BOM DIA e a string2 – bom dia é diferente pois o valor inteiro Unicode de uma letra minúscula é diferente do valor para letra maiúscula

Comparando Strings

- O método `CompareTo` retorna
 - 0 se as strings são iguais
 - 1 se a string que invoca o método é menor que a string passada como argumento
 - 1 se a string que invoca o método é maior que a string passada como argumento

Comparando Strings

- CompareTo

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1 = "BOM DIA";
        string string2 = "bom dia";
        int resultado;
        Console.WriteLine ("String 1: {0} - String 2: {1}" , string1, string2);
        //testa a igualdade usando CompareTo
        resultado = string1.CompareTo (string2);
        Console.WriteLine ("A comparacao entre a string 1 e 2 eh:" + resultado);
    }
}
```

- Saída

```
String 1: BOM DIA - String 2: bom dia  
A comparacao entre a string 1 e 2 eh:1  
Press any key to continue...
```

Comparando Strings

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1 = "BOA TARDE";
        string string2 = "BOM DIA";
        int resultado;
        Console.WriteLine ("String 1: {0} - String 2: {1}" , string1, string2);

        resultado = string1.CompareTo (string2);
        Console.WriteLine ("A comparacao entre a string 1 e 2 eh:" +
resultado);
    }
}
```

- Saída

```
String 1: BOA TARDE - String 2: BOM DIA  
A comparacao entre a string 1 e 2 eh:-1
```

```
Press any key to continue...
```


Comparando Strings

```
class ConstroiString
{
    public static void Main (string[] args)
    {
        string string1 = "BOA TARDE";
        string string2 = "BOA TARDE";
        int resultado;
        Console.WriteLine ("String 1: {0} - String 2: {1}" , string1, string2);

        resultado = string1.CompareTo (string2);
        Console.WriteLine ("A comparacao entre a string 1 e 2 eh:" +
resultado);
    }
}
```

- Saída

```
String 1: BOA TARDE - String 2: BOA TARDE
```

```
A comparacao entre a string 1 e 2 eh:0
```

```
Press any key to continue...
```

Comparando Strings

- O método ***StartsWith*** determina se uma instância de uma string começa com um texto passado como argumento
- E o método ***EndsWith*** determina se uma instância de uma string termina com um texto passado como argumento

Comparando Strings

```
class ConstroiString
{
    public static void Main (string[] args)
    {

        string [ ] strings = { "inicio", "inicializar", "meio", "fim"} ;
        for (int i =0; i < strings.Length; i++)
        {
            if (strings[i].StartsWith ("ini"))
                Console.WriteLine("A string {0} comeca com ini", strings[i]);
        }
        for (int i =0; i < strings.Length; i++)
        {
            if (strings[i].EndsWith ("eio"))
                Console.WriteLine("A string {0} termina com eio", strings[i]);
        }
    }
}
```

- Saída

```
A string inicio comeca com ini
A string inicializar comeca com ini
A string meio termina com eio

Press any key to continue...
```

Localizando caracteres e substrings

- O método ***IndexOf*** localiza a primeira ocorrência de um caracter ou substring em uma string
- Se o método encontra um caracter, ele retorna o índice do caracter especificado. Caso contrário, o método retorna -1

Localizando caracteres e substrings

- O método ***LastIndexOf*** localiza a última ocorrência de um caracter em uma string
- Se o caracter é encontrado, o seu índice é retornado; caso contrário o método retorna -1

Localizando caracteres e substrings

```
public static void Main (string[] args)
{
    string letras = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnop";
    int posicao;
    posicao = letras.IndexOf('c');
    Console.WriteLine("c esta na posicao: " + posicao);

    posicao = letras.IndexOf('a', 1);
    Console.WriteLine("a esta na posicao: " + posicao);

    posicao = letras.IndexOf('$', 3, 5);
    Console.WriteLine("$ esta na posicao: " + posicao);

    posicao = letras.LastIndexOf('c');
    Console.WriteLine("A ultima ocorrencia de c esta na posicao: " + posicao);

    posicao = letras.LastIndexOf('a', 25);
    Console.WriteLine("A ultima ocorrencia de a esta na posicao: " + posicao);

    posicao = letras.LastIndexOf('$', 15, 5);
    Console.WriteLine("A ultima ocorrencia de $ esta na posicao: " + posicao);
}
```


- Saída

```
c esta na posicao: 2
a esta na posicao: 26
$ esta na posicao: -1
A ultima ocorrencia de c esta na posicao: 28
A ultima ocorrencia de a esta na posicao: 0
A ultima ocorrencia de $ esta na posicao: -1

Press any key to continue...
```

Localizando caracteres e substrings

- `posicao = letras.IndexOf('a', 1)`
 - Dois argumentos: o caracter a ser procurado e o índice inicial onde a busca vai começar. O método não examina quaisquer caracteres antes do índice inicial
- `posicao = letras.IndexOf('$', 3, 5)`
 - Três argumentos: o carácter a ser procurado, o índice no qual a pesquisa vai começar e a quantidade de caracteres a serem pesquisados

- `posicao = letras.LastIndexOf('c');`
 - Recebe como argumento o caracter a ser procurado
- `posicao = letras.LastIndexOf('a', 25);`
 - Dois argumentos: o caracter a ser procurado e o índice mais alto a partir do qual vai começar a busca para trás do caracter
- `posicao = letras.LastIndexOf('$', 15, 5);`
 - Três argumentos: o carácter a ser procurado, o índice mais alto a partir do qual a busca para trás vai começar e o número de caracteres a serem pesquisados

Localizando caracteres e substrings

```
public static void Main (string[] args)
{
    string letras = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnop" ;
    int posicao;
    posicao = letras.IndexOf("cde");
    Console.WriteLine("cde esta na posicao: " + posicao);

    posicao = letras.IndexOf("abc", 1);
    Console.WriteLine("abc esta na posicao: " + posicao);

    posicao = letras.IndexOf("$ef", 3, 5);
    Console.WriteLine("$ef esta na posicao: " + posicao);

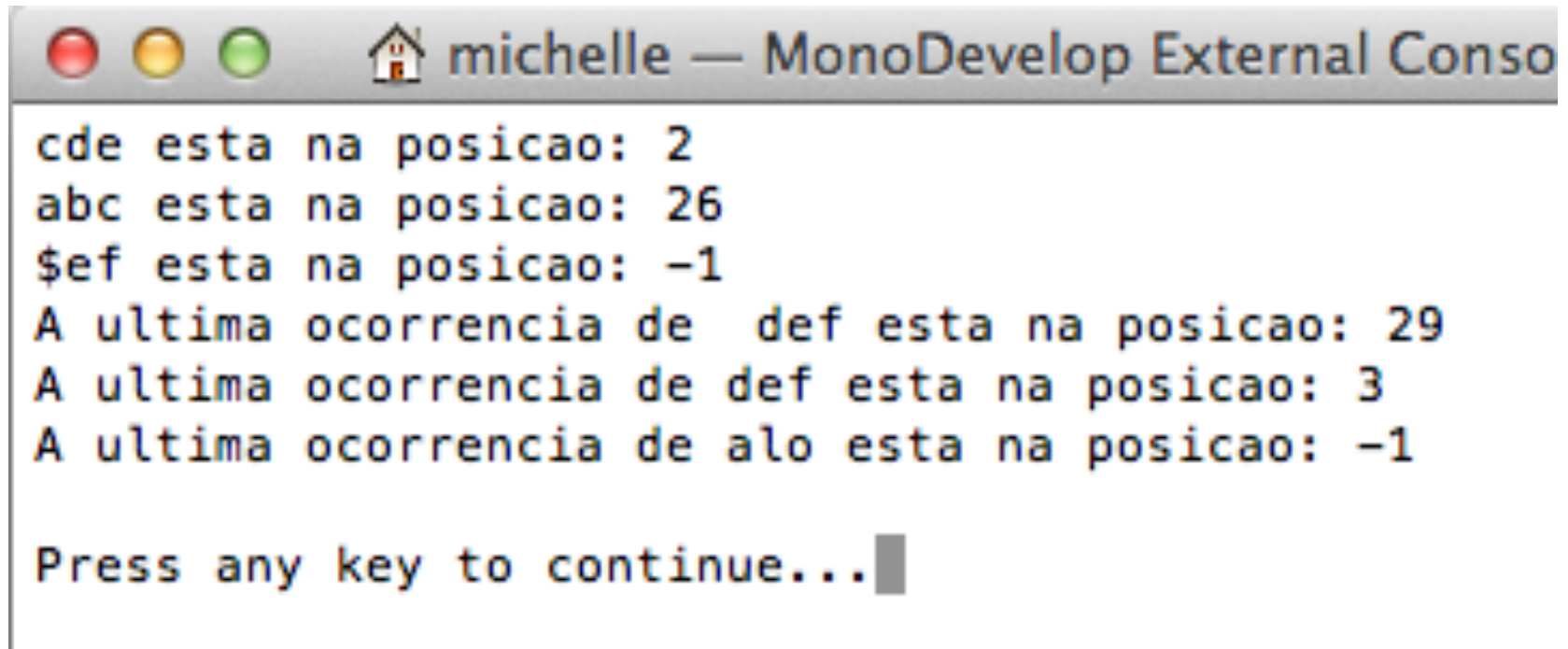
    posicao = letras.LastIndexOf("def");
    Console.WriteLine("A ultima ocorrencia de def esta na posicao: " + posicao);

    posicao = letras.LastIndexOf("def" , 25);
    Console.WriteLine("A ultima ocorrencia de def esta na posicao: " + posicao);

    posicao = letras.LastIndexOf("alo", 20, 15);
    Console.WriteLine("A ultima ocorrencia de alo esta na posicao: " + posicao);
}
```

- As versões de *IndexOf* e *LastIndexOf* recebem uma string, em vez de um caracter, como primeiro argumento
- Essas versões dos métodos funcionam do mesmo modo que as anteriores, exceto que retornam o índice da primeira ocorrência de qualquer um dos caracteres

- Saída:



The image shows a screenshot of a MonoDevelop External Console window. The title bar at the top reads "michelle — MonoDevelop External Conso". The console contains the following text:

```
cde esta na posicao: 2  
abc esta na posicao: 26  
$ef esta na posicao: -1  
A ultima ocorrencia de  def esta na posicao: 29  
A ultima ocorrencia de def esta na posicao: 3  
A ultima ocorrencia de alo esta na posicao: -1  
  
Press any key to continue...
```

Extraindo substrings de strings

```
public static void Main (string[] args)
```

```
{
```

```
    string letras = "abcdefghijklmnopqrstuvwxyzabcdefghijklmenop" ;
```

```
    string resultado;
```

```
    resultado = letras.Substring(20);
```

```
    Console.WriteLine("\n" + resultado);
```

```
    resultado = letras.Substring(0, 6);
```

```
    Console.WriteLine("\n" + resultado);
```

```
}
```

uvwxyzabcdefghijklmenop

abcdef

Press any key to continue

Saída:

Extraindo substrings de strings

- resultado = letras.Substring(20);
 - O argumento no método Substring especifica o índice a partir do qual o método copia caracteres da string original até o fim da mesma
- resultado = letras.Substring(0, 6);
 - O primeiro argumento especifica o índice a partir do qual o método copia caracteres da string original e o segundo argumento especifica o comprimento da substring a ser copiada

Concatenando strings

- Podemos concatenar strings usando o operador (+) e também o método Concat

```
public static void Main (string[] args)
{
    string string1 = "Feliz ";
    string string2 = "Aniversario";
    Console.WriteLine("Concatenando com +: " +string1 + string2);
    Console.WriteLine("Concatenando com Concat:" +string.Concat(string1,
string2));
}
```

- Saída

```
Concatenando com +: Feliz Aniversario
Concatenando com Concat:Feliz Aniversario

Press any key to continue... █
```

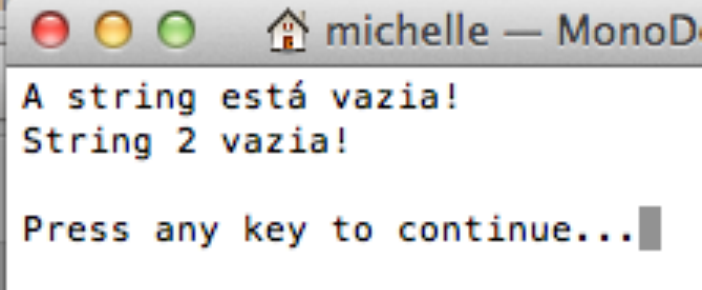
- Como dito anteriormente, a palavra-chave `string` é um apelido de C# para a classe `System.String`
- Dessa forma, quando um `string` é criado, um objeto da classe `System.String` é criado

Campos da classe String

- Empty
 - Representa um string vazio (" "), de comprimento zero

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "";
            string str2 = String.Empty;
            if (str1 == String.Empty)
                Console.WriteLine ("A string está vazia!");
            else
                Console.WriteLine("String nao vazia!");
            if (str2 == "")
                Console.WriteLine("String 2 vazia!");
            else
                Console.WriteLine("String 2 nao vazia!");
        }
    }
}
```



michelle — MonoD

A string está vazia!
String 2 vazia!

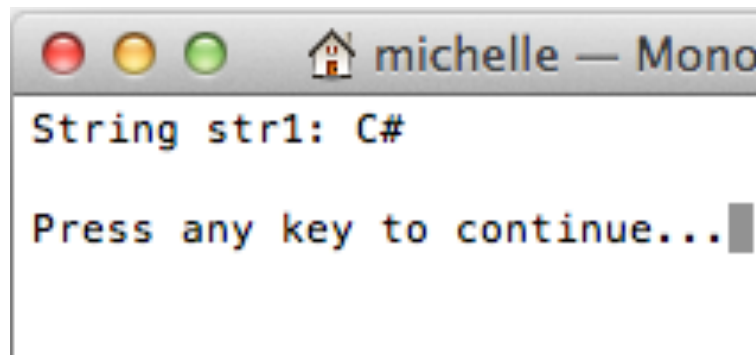
Press any key to continue...

Construtores da classe String

- Cria um string a partir de um array de caracteres
 - `public String (char[] c)`

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            char [] c = {'C', '#'};
            string str1 = new string(c);
            Console.WriteLine("String str1: " + str1);
        }
    }
}
```



Construtores da classe String

- Cria um string com n caracteres do array de caracteres c, iniciando na posição de índice início. Se n for zero, será criado um string vazio
 - `public String (char [] c, int início, int n)`

Construtores da classe String

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            char [] c = {'D', 'I', 'G', 'I', 'T', 'A', 'L'};
            string str1 = new string(c, 2, 4);
            Console.WriteLine("String str1: " + str1);
        }
    }
}
```

 michelle — MonoD

String str1: GITA

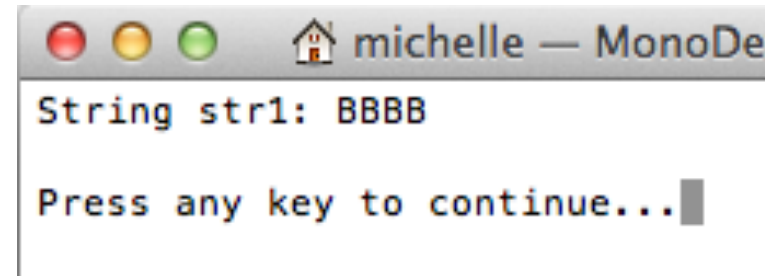
Press any key to continue...

Construtores da classe String

- Cria um string contendo o caractere c, n vezes.
Se n for zero, será criado um string vazio
 - `public String (char c, int n)`

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = new string('B', 4);
            Console.WriteLine("String str1: " + str1);
        }
    }
}
```



The screenshot shows a MonoDevelop window titled "michelle — MonoDe". The console output displays "String str1: BBBB" followed by a prompt "Press any key to continue..." with a cursor.

- Concat
 - Retorna um string que é a concatenação de strings

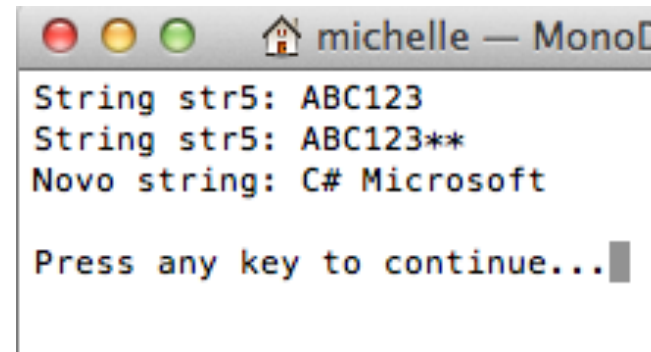
```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ABC";
            string str2 = "123";
            string str3 = "**";
            string [] str4 = {"C# ", "Micro", "soft"};

            string str5 = String.Concat(str1, str2);
            Console.WriteLine("String str5: " + str5);

            str5 = String.Concat(str1, str2, str3);
            Console.WriteLine("String str5: " +str5);

            string str6 = String.Concat(str4);
            Console.WriteLine("Novo string: " +str6);
        }
    }
}
```



String str5: ABC123
String str5: ABC123**
Novo string: C# Microsoft
Press any key to continue...

Métodos da Classe String

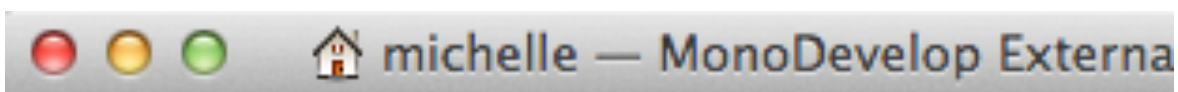
- Join()
 - Concatena um separador entre cada elemento do array de strings, retornando um único string

```
public static string Join(string separador, string [ ] str)
```

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string [] nomes = {"Carla ", "Paula", "Ana", "Marcela"};

            string separador = " - ";
            string novastring;
            novastring = String.Join (separador, nomes);
            Console.WriteLine("Novo string: " +novastring);
        }
    }
}
```



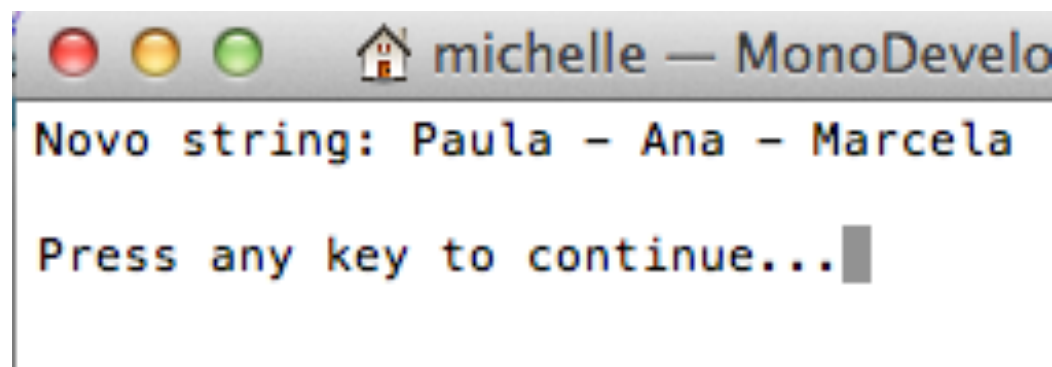
Novo string: Carla - Paula - Ana - Marcela

Press any key to continue...█

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string [] nomes = {"Carla ", "Paula", "Ana", "Marcela", "Adriana"};

            string separador = " - ";
            string novastring;
            novastring = String.Join (separador, nomes, 1, 3);
            Console.WriteLine("Novo string: " +novastring);
        }
    }
}
```



Métodos da Classe String





- `ToLower()`
 - Converte o string chamador em letras minúsculas
- `ToUpper()`
 - Converte o string chamador em letras maiúsculas

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string [ ] str1 = {"CARLA ", "ANA", "MARCELA", "LUISA"};
            string [] str2 = new string[str1.Length];

            for (int i =0; i < str1.Length; i++) {
                str2 [i] = str1[i].ToLower ();
                Console.WriteLine ("Novo string minuscuro: " + str2[i]);
            }

            Console.WriteLine("\n\n");
            string [] str3 = new string[str2.Length];
            for (int j=0; j< str2.Length; j++)
            {
                str3[j] = str2[j].ToUpper();
                Console.WriteLine ("Novo string maiusculo: " + str3[j]);
            }
        }
    }
}
```

    michelle — MonoDevelo

```
Novo string minusculo: carla  
Novo string minusculo: ana  
Novo string minusculo: marcela  
Novo string minusculo: luisa
```

```
Novo string maiusculo: CARLA  
Novo string maiusculo: ANA  
Novo string maiusculo: MARCELA  
Novo string maiusculo: LUISA
```

```
Press any key to continue...
```

Métodos da Classe String

- `ToCharArray()`
 - Copias os caracteres do string chamador para um array de caracteres
 - Se o string chamador for vazio, o array retornado será vazio

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ALGORITMOS";
            char [] palavra;

            palavra = str1.ToCharArray();

            for (int i =0; i < palavra.Length; i++) {
                Console.Write(palavra[i] + " ");
            }
        }
    }
}
```

A L G O R I T M O S

Press any key to continue...


```

using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ABCDEFGHIJKLM";
            char [] palavra;

            palavra = str1.ToCharArray(3,3);
            Console.Write("As letras de {0} sao: ", str1);
            Console.Write(palavra);
            Console.WriteLine("\n");
            Console.Write ("Cada letra de {0}: \n| ", str1);
            foreach(char i in palavra)
                Console.Write(i);
        }
    }
}

```

As letras de ABCDEFGHIJKLM sao: DEF

Cada letra de ABCDEFGHIJKLM:

DEF


Press any key to continue...

Métodos da Classe String

- Remove()
 - Elimina caracteres do string chamador a partir da posição de índice início até a última posição, retornando o último string

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ABCD####ABCD", str2;
            str2 = str1.Remove (4);
            Console.WriteLine (str2);
        }
    }
}
```

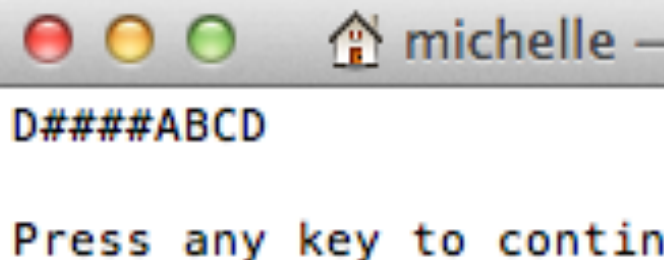
 michelle — Mono

ABCD

Press any key to continue...

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ABCD####ABCD", str2;
            str2 = str1.Remove (0, 3);
            Console.WriteLine (str2);
        }
    }
}
```



D####ABCD

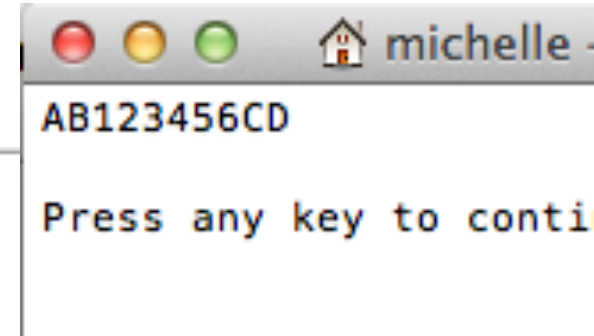
Press any key to continue...

- Insert()
 - Insere no string na posição de índice início do string chamador

public string Insert(int início, string str)

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "ABCD", str2 = "123456", str3;
            str3 = str1.Insert (2, str2);
            Console.WriteLine (str3);
        }
    }
}
```



- Split()

```
using System;

namespace Strings2
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            string str1 = "Um,Dois,Tres,Quatro" ;
            string [] substring;
            char []separador = {','};
            substring = str1.Split (separador);
            foreach(string s in substring)
                Console.WriteLine (s);
        }
    }
}
```

Um
Dois
Tres
Quatro

Press any key to